

به نام خدا

گزارش تمرین اول درس هوش مصنوعی

آریا کوثری ۹۴۱۱۰۱۴۷

امیررضا مزینی ۹۴۱۱۰۱۶۹

علیرضا وزوایی ۹۴۱۰۴۳۴۸

الگوریتم

برای حل این مساله از الگوریتمی مشابه hill climbing استفاده شده است.

در فضای حل مساله هر سلکتور معادل یک حالت است و همسایه های هر سلکتور، سلکتورهایی هستند که با اضافه کردن "nth-of-type" به سلکتور آن به دست آمده اند. به عبارتی دیگر همسایه های هر سلکتور، سلکتورهای تگ های درونی تگ مربوط به آن سلکتور هستند.

با بحث و تبادل نظر های انجام شده بین اعضای گروه، نتیجه گیری شد که حالت اولیه تصادفی برای این مساله مناسب نیست (زیرا بهتر است هنگام بررسی یک سلکتور ابتدا سلکتور والدش بررسی شده باشد) و به این ترتیب سلکتور "body" به عنوان حالت اولیه انتخاب شد.

امتیاز هر سلکتور برابر تعداد محتواهای مورد نظری است که شامل می شود. این الگوریتم اگر امتیاز سلکتور فعلی با امتیاز سلکتور همسایه اش برابر باشد به سلکتور همسایه می رود زیرا همیشه سلکتور های همسایه خاص تر هستند و اگر امتیاز سلکتور همسایه از امتیاز سلکتور فعلی کمتر باشد سلکتور فعلی را خروجی می دهد زیرا می خواهد سلکتوری بدهد که تمامی محتواها را شامل شود.

علاوه بر "nth-of-type" برای ساده تر شدن متن سلکتورها در صورت امکان از id و class تگ ها نیز استفاده شد.

معماری سیستم

سیستم با زبان برنامه نویسی پایتون پیاده سازی شده و برای تحلیل محتوای اینترنتی از پکیج BeautifulSoup استفاده شده است.

سیستم حاوی ۳ فایل get_input.py و main.py و searcher.py است که شرح هرکدام در زیر آمده است:

- `get-input.py`: برای آغاز به کار سیستم باید این فایل اجرا شود. در اینجا لینک و محتوا های ورودی از فایل `input.json` خوانده می شوند و محتوای هر لینک از اینترنت دریافت می شود و به تابع `parser_agent` از فایل `main.py` ورودی داده می شود.
- `main.py`: بخش اصلی الگوریتم در تابع `parser_agent` این فایل انجام شود. این تابع سلکتور فعلی (که در ابتدا `body` است) و محتواها را ورودی میگیرد و در ابتدا همسایه های سلکتور فعلی را پیدا می کند و سپس به کمک تابع `search` از فایل `searcher.py` برای هر همسایه تعداد محتواهایی که در تگ مربوط به سلکتورش هستند را پیدا میکند. اگر برای همه همسایه ها این تعداد از تعداد کل کمتر بود همین سلکتور را خروجی داده می شود و در غیر این صورت تابع به صورت بازگشتی برای سلکتوری که تعداد محتوا را حفظ میکند صدا زده می شود. همچنین در این تابع برای ساده کردن سلکتور در صورت امکان از `id` و `class` تگ مربوطه استفاده می شود.
- `searcher.py`: حاوی تابع `search` است که یک سلکتور و محتوا را ورودی میگیرد و بررسی میکند آیا آن سلکتور آن محتوا را شامل می شود یا نه.

روند پیاده سازی سیستم در مخزن گیت در آدرس https://github.com/somagh/AI_HW1 قابل مشاهده است.

نحوه ران کردن برنامه

- 1- مفسر پایتون ۳ را بر روی سیستم عامل خود نصب کنید.
- 2- پکیج `BeautifulSoup4` را بر روی پایتون خود نصب کنید.
- 3- سه فایل `get_input.py` و `main.py` و `searcher.py` در یک دایرکتوری قرار دهید.
- 4- در `cmd` یا ترمینال خود به دایرکتوری ذکر شده بروید و فرمان `"python get_input.py"` را بزنید.
- 5- آدرس فایل ورودی را وارد کنید.
- 6- سلکتور به دست آمده در ادامه چاپ خواهد شد.